УДК 681.3

О.М. Федорус
Київський національний університет імені Тараса Шевченка, Україна
вул. Володимирська 60, м. Київ, 01033

ПАРКС ЯК ЗАСІБ РЕАЛІЗАЦІЇ РОЗПОДІЛЕНИХ ХМАРНИХ ОБЧИСЛЕНЬ

O.M. Fedorus
Taras Shevchenko National University of Kyiv, Ukraine
60 Volodymyrska Street, City of Kyiv, Ukraine, 01033

PARCS as a tool for implementing distributed cloud computing

У статті викладений огляд технології ПАРКС, у якому відзначаються основні особливості, що дозволяють використовувати ПАРКС як засіб реалізації розподілених хмарних обчислень.
Ключові слова: ПАРКС, хмарні обчислення, розподілені обчислення.
In the article the review PARCS technology, which marked the main features that allow you to use PARCS as a means of implementing distributed cloud computing.
Keywords: PARCS, Cloud computing, distributed computation.

Вступ

Зі швидким розвитком технологій обробки та зберігання інформації та успіхом Інтернету обчислювальні ресурси стали дешевші, потужніші та більш розповсюджені, ніж коли-небудь [1]. Такий розвиток дозволив реалізацію так званих хмарних обчислень, у яких ресурси (наприклад, CPU, RAM та простір на диску) надаються як послуги і можуть бути збільшені, або зменшені, на вимогу користувача. Технології хмарних обчислень зробили значний вклад у розвиток IT індустрії. Майже всі гіганти IT індустрії так чи інакше займаються хмарними технологіями. Google, Microsoft та Amazon змагаються між собою за можливість надати більш потужні, надійні та ефективні хмарні платформи.

До хмарних обчислень відносять програмне забезпечення як сервісів, що надаються через інтернет та апаратні і системні засоби в центрах обробки даних, що надають ці сервіси. Ці сервіси вже давно називають «Програмне забезпечення як послуга» (SaaS). Апаратні засоби датацентру та програмні засоби - це те, що ми називаємо хмарою.

Хмарні обчислення мають декілька переваг, що роблять їх привабливими для бізнесу:

1) *Нема інвестицій «у майбутнє».* Користувачам не потрібно інвестувати в інфраструктуру. Вони просто орендують ресурси в хмарі відповідно до своїх потреб.
2) *Менші витрати на користування.* Об'єм ресурсів у хмарі може бути швидко змінений на вимогу користувача. А отже, нема потреби постійно використовувати ресурси, які потрібні для пікового навантаження. Це дає можливість економії, коли навантаження на сервіс мінімальне.
3) *Легко масштабується.* Провайдери інфраструктури мають великий пул ресурсів у датацентрах і роблять його легкодоступним. А сервіс-провайдери, у свою чергу, можуть легко розширити свій сервіс до великих масштабів при потребі. Наприклад, при різкому зростанні потреб сервісу.
4) *Легкий доступ.* Сервіси, що розміщуються у хмарі загалом є web-сервісами, а отже, вони легкодоступні на будь-яких пристроях, які мають зв'язок з інтернетом. Це не тільки звичайні комп'ютери та ноутбуки, а й телефони і смартфони.

5) *Зменшують ризики та витрати на підтримку для бізнесу.* Усі ризики, зв'язані з відмовою інфраструктурних об'єктів, чи витрати на їх ремонт перекладаються на провайдерів інфраструктури.

Зрозуміло, що такі переваги стали доступні завдяки «віртуальності» наданих послуг. Обчислення, що проводяться у хмарі, не прив'язані до конкретних апаратних ресурсів, які можуть змінюватись відповідно до об'ємів обчислень та загальних потреб. З цього приводу цікаво розглянути можливість застосування системи ПАРКС для хмарних обчислень, оскільки вона містить у собі концепції, які дозволяють вести мову про можливість і ефективність реалізації таких систем за допомогою ПАРКС.

ПАРКС

ПАРКС-технологія програмування представляє собою деяку множину програмних засобів, що забезпечують процес розробки і реалізації алгоритмів паралельної обробки інформації [2,3] і базується на концепції керуючого простору (КП), запропонованої в праці [5], і який представляється віртуально паралельним простором (ВПП) [4]. Керуючий простір є динамічним графом, який використовується для опису логічної та комунікаційної структури досліджуваної задачі і відображає динамічні зміни в ній.

Завдяки керуючому простору ПАРКС- технологія дозволяє:

1) підтримувати структурне паралельне програмування на основі статичного та динамічного паралелізму;

2) використовувати різноманітні засоби опису рекурсії по даних і рекурсії по керуванню;

3) на логічному рівні в явному вигляді описувати розподілення ресурсів, схеми комутації і перекомутації зв'язків або отримувати логічну структуру системи, розподілення ресурсів і схему перекомутації як результат обробки і побудови керуючого простору;

4) будувати систему віртуальних ПАРКС-машин, що забезпечують можливість паралельної обробки інформації: ефективна реалізація системи віртуальних ПАРКС-машин можлива лише на широкому класі паралельних і спеціалізованих ЕОМ.

Основними термінами ПАРКС- технології є: точка, програмний канал (ПК), алгоритмічний модуль (АМ).

Структура керуючого простору – граф, вершини якого – точки керуючого простору, а ребра – програмні канали, які їх з'єднують. При цьому одні й ті самі точки можуть бути з'єднані за допомогою декількох програмних каналів різного типу. До кожної точки керуючого простору приписаний алгоритмічний модуль, який є процедурою ПАРКС-розширення базової мови. Виконання алгоритмічного модуля може призвести до зміни у структурі керуючого простору: видаленню та створенню точок, програмних каналів, блокуванню виконання деяких процесів і т.д. Алгоритмічні модулі виконуються паралельно, однак, модуль може бути затриманий до приходу деякої події або на деякий час. Точки керуючого простору можуть зберігати в собі керуючий простір довільної глибини, керуючий простір може мати деякий рівень рекурсії. Точки керуючого простору, з'єднані за допомогою програмного каналу, можуть посилати і получати повідомлення.

Алгоритмічний модуль є процедурою ПАРКС-розширення базової алгоритмічної мови програмування (наприклад, Java, Pascal, C#, C). Це розширення повинно забезпечувати для віртуальних ПАРКС-машин рівні можливості [3].

1) Програмні засоби, що забезпечують побудову і модифікацію керуючого простору. Керуючий простір складається з точок і каналів, що з'єднують точки. У

кожній точці на наступному рівні ієрархічного представлення може розміщуватись новий керуючий простір. При побудові і обробці керуючого простору допустимо використання рекурсії по структурі керуючого простору. Точка керуючого простору однозначно визначається деяким логічним номером. Канали керуючого простору призначені для забезпечення взаємозв'язків між точками. Вони також типізовані. Зокрема, типи каналів задають ієрархічну структуру керуючого простору. Для кожного алгоритму існує відповідна йому структура керуючого простору, що відображає його логічну структуру. Наприклад, для паралельного алгоритму перемноження матриць великої розмірності керуючий простір буде представлено 8-зв'язним графом [5].

Програмні засоби цього рівня повинні забезпечувати наступні можливості: створювати нову точку керуючого простору заданого типу та визначати для неї новий логічний номер; з'єднати точки керуючого простору каналом заданого типу; визначити, які точки керуючого простору під'єднанні до заданої точки і через які канали; знищити точку керуючого простору і всі канали, що зв'язують її з іншими точками; знищити канали заданого типу, що з'єднують точки керуючого простору; визначити логічний номер і тип точки керуючого простору.

2) Програмні засоби, що забезпечують зберігання та передачу інформації за допомогою керуючого простору. На цьому рівні виділяються нумеровані канали керуючого простору, через які уточнюється структура взаємодії між точками КП.

Програмні засоби цього рівня мусять забезпечувати наступні можливості системи: створювати кінцеве число моніторів точки керуючого простору, що задають правила зберігання, отримання і передачі інформації (на основі базової алгоритмічної мови); створювати блок інформації; записувати в канал із заданим номером блок інформації; перевіряти існування блоків інформації в каналі з заданим номером; прочитати інформацію з каналу з заданим номером блоку інформації; чекати запису блоку інформації в будь-який з каналів із заданим номером.

3) Програмні засоби, що забезпечують роботу з алгоритмічними модулями. Алгоритмічній модуль призначений для уточнення одиниці паралельної роботи базової алгоритмічної мови. За допомогою алгоритмічного модуля здійснюється побудова, модифікація та наповнення керуючого простору.

ПАРКС-модель визначається наступним шляхом:

1) Виділяються кінцеві набори базових алгоритмічних модулів, що описують шляхи розпаралелювання алгоритмів по різних рівнях деталізації.
2) Визначають правила побудови, видалення і взаємодії алгоритмічного модулю на усіх рівнях деталізації.
3) Уточнюється поняття рекурсії по даних і рекурсії по керуванню для заданої базової алгоритмічної мови.

Процес функціонування ПАРКС-моделі складає створення та знищення активних копій алгоритмічних модулів, їх частково децентралізоване асинхронне паралельне функціонування та динамічну взаємодію цих модулів з можливістю створення активних копій алгоритмічного модуля.

Опис алгоритмічного модуля представляє собою розширення відповідної конструкції базової мови операціями віртуальних рівнів 1-3.

Програмні засоби рівня 3 повинні забезпечувати наступні можливості:

1) Створювати, видаляти та модифікувати опис алгоритмічного модуля.
2) Створювати активну копію алгоритмічного модуля і приписувати її для виконання у точці керуючого простору.

3) Відсилати повідомлення активній копії алгоритмічного модуля (відкривати доступ до розділених структур даних).
4) Приймати повідомлення від активної копії алгоритмічного модуля (закрити доступ до розділених структур даних).
5) Визначити головну активну копію алгоритмічного модуля (зі створення якої починається функціонування ПАРКС-системи).
6) Здійснювати вибіркове очікування та виконання активної копії алгоритмічного модуля.

Рекурсія на рівні алгоритмічного модуля проявиться в тому, що активні копії алгоритмічного модуля можуть створювати нові активні копії по алгоритмічному модулю. Апарат створення, знищення, модифікації, приписки активної копії алгоритмічного модуля в точки керуючого простору, взаємодії активних копій алгоритмічного модуля дозволяє описувати рекурсію по даних, рекурсію по керуванню і їх композицію.

Розробка ПАРКС-моделі передбачає можливість автоматичної побудови керуючого простору, що відображає динамічну структуру взаємозв'язків у ПАРКС-моделі, передбачає існування необмеженого паралелізму.

Використовуючи програмні засоби рівнів 1 та 2, можна враховувати ряд обмежень, що накладаються на структуру розроблюваної системи, планувати розподілення ресурсів, забезпечувати ефективне використання обмежених ресурсів і перекомутацію зв'язків. Для рівня 3 можливе введення засобів роботи з часом и засобів, що забезпечують моделювання.

4) Програмні засоби, що забезпечують процедурно-об'єктно-орієнтоване програмування. Технологія процедурно-об'єктно-орієнтованого програмування визначається і обмежується базовою алгоритмічною мовою.

Обирається програмна конструкція базової мови. Наприклад, для мови Ада – механізм пакетів та задач з уніфікованим викликом процедур пакетів і точок входів задач, для мов С та Java – набір класів з описом відповідних методів і т.д. Обрана конструкція розглядається як опис алгоритмічного модуля. Уніфікований виклик процедур і точок входу транслюється на рівень 3, де розписується за допомогою програмних засобів, що забезпечують взаємодію активних копій алгоритмічних модулів.

Реалізація програмних засобів рівнів 1 і 2 залежить від архітектури ЕОМ та її операційної системи, а рівнів 3 та 4 від базової алгоритмічної мови програмування.

На наступних рівнях розроблюються програмні засоби, використання яких забезпечує автоматичну генерацію паралельних програм по їх специфікації.

Таким чином, система програмування ПАРКС – сукупність програмних засобів, які підтримують процес розробки і реалізації алгоритмів паралельної обробки інформації за рахунок розширення базових алгоритмічних мов (C, Pascal, FORTRAN, Modula-2, Java) [6]. Характерною особливістю системи програмування ПАРКС є те, що запропоновані нею засоби - найбільш потужні алгоритмічні концепції: рекурсія та паралельність, призначені для опису взаємодій зі змінною комутацією зв'язків і рекурсивно-паралельним розвитком процесів, описаних на основі алгоритмічної мови. Акцент робиться на програмні засоби опису динамічного паралелізму.

Іншим важливим критерієм класифікації систем паралельної обробки, крім статичності/динамічності, можна вважати відкритість або закритість програмного забезпечення. Ряд розробок і технологій (наприклад, Ада, OCCAM, МАЯК та ін.) пропонують як базовим заново створену мову (як правило, складну), примушуючи користувача

вивчати її. Це - позиція закритих систем і технологій. Сильною стороною такого підходу є можливість досягти високої ефективності виконання для деяких класів задач.

Альтернативні відкриті системи і технології (наприклад, MANIFOLD, LINDA, MPICH, OpenMP та ін.) надають користувачам можливість працювати у найбільш зручній для його задач мові (як базової), доповнюючи його тим чи іншим засобом паралелізму. Крім комфортності для користувача, у відкритих системах часто спрощуються і синтаксичні проблеми, оскільки вони здебільшого концентруються у паралелізмі. Можна з впевненістю заявити, що в рамках однієї, нехай і ретельно розробленої, мови програмування не вдасться задовольнити усі категорії користувачів і вирішити усі проблеми [6]. Більш логічним, на думку авторів [6], є використання базового набору програмних засобів, які дозволяють у рамках широкого класу мов працювати на деякій технології. ПАРКС-система програмування підтримує ПАРКС-технологію програмування, яка базується саме на цій ідеї. Змінюючи базові алгоритмічні мови, можна отримати сукупність програмних засобів, які охоплюють широкий спектр – від мов проектування спецпроцесорів до універсальних мов програмування. У цьому і полягає оригінальна особливість ПАРКС-технології.

Зрозуміло, що при побудові КП (ВПП) на мережі, основну роль відіграє процес, який забезпечує побудову КП: створює точки, канали, приписує АМ точкам, підтримує їх взаємодію. Умовна назва – демон. Обмін даними з ним проходить за допомогою TCP/IP, або через локальний файл.

Висновки

ПАРКС технологія завдяки концепціям КП, рекурсивності та необмеженого паралелізму може бути застосована при організації хмарних обчислень. Як вже було описано у вступі, «хмарні обчислення» характеризуються двома складовими частинами: апаратно-системні засоби та сервіс доступу до них для користувачів. Зазначимо, як вони можуть бути реалізовані (або позначені) у термінах ПАРКС.

Апаратно-системні засоби. Будуть представлятися комп'ютерами, об'єднаними у локальну мережу, з запущеними на них службами-демонами і на які можуть бути використані для реалізації структури КП.

Сервіс доступу. Деякий програмний засіб, що дозволяє запустити ПАРКС-програму на наявних апаратно-системних засобах. Також, сюди належать файли, необхідні для створення ПАРКС-програми.

Також, будь-який засіб хмарних обчислень має реалізовувати масштабування. Масштабування у ПАРКС-системах реалізується завдяки тому, що КП є віртуальним і може бути розміщений як на одній ЕОМ, так і на мережі (або підмережі). А завдяки рекурсивності КП, може бути розширений під час виконання ПАРКС-програми.

Література

1. Qi Zhang, Lu Cheng, Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. //The Brazilian Computer Society 2010 – 20 April 2010 p. 7-18.
2. Анисимов А.В., Кулябко П.П. Программирование параллельных процессов в управляющих пространствах. // Кибернетика 1984 - №3 – С 79-88.
3. Анисимов А.В., Кулябко П.П. Особенности ПАРУС-технологии // Кибернетика и системный анализ – 1993 - № 3 – С. 128-137.
4. Кулябко П.П., В.П.Шевченко, Анісімова Л.А., Ветров А.М. Керуючий простір як віртуально паралельний простір. Вісник Київського університету серія: фіз.-мат. наук, 1999. №4 с. 82-86.
5. Глушков В.М., Анисимов А.В. Управляющие пространства в асинхронных параллельных вычислениях // Кибернетика – 1980 - №5 – с. 1-9.
6. Анисимов А.В., Деревянченко А.В. Система ПАРУС-Java для параллельных вычислений на компьютерных сетях // Кибернетика и системный анализ -2005- №1 – с. 25-36.

Literatura

1. Qi Zhang, Lu Cheng, Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. //The Brazilian Computer Society 2010 – 20 April 2010 p. 7-18.
2. Anisimov A.V., Kulyabko P.P. Programmirovanie paralelnyih protsessov v upravlyayuschih prostranstvah. // Kibernetika 1984 - #3 – S 79-88.
3. Anisimov A.V., Kulyabko P.P. Osobennosti PARUS-tehnologii // Kibernetika i sistemnyiy analiz – 1993 - # 3 – S. 128-137.
4. Kulyabko P.P., V.P.Shevchenko, AnIsImova L.A., Vetrov A.M. Keruyuchiy prostIr yak vIrtualno paralelniy prostIr. VIsnik kiYivskogo unIversitetu serIya: fIz.-mat. nauk, 1999. #4 s. 82-86.
5. Glushkov V.M., Anisimov A.V. Upravlyayuschie prostranstva v asinhronnyih parallelnyih vyichisleniyah // Kibernetika – 1980 - #5 – s. 1-9.
6. Anisimov A.V., Derevyanchenko A.V., Sistema PARUS-Java dlya parallelnyih vyichisleniy na kompyuternyih setyah // Kibernetika i sistemnyiy analiz -2005- #1 – s. 25-36.

RESUME

O.M. Fedorus

PARCS as a tool for implementing distributed cloud computing

The emergence of cloud computing has made a tremendous impact on the Information Technology (IT) industry over the past few years, where large companies such as Google, Amazon and Microsoft strive to provide more powerful, reliable and cost-efficient cloud platforms, and business enterprises seek to reshape their business models to gain benefit from this new paradigm.

These advantages have become available thanks to virtuality of these services. Computations carried out in the cloud are not tied to specific hardware resources, which may vary according to the volume and general computing needs. In this respect interesting to consider the possibility of PARCS-systems for cloud computing, because it contains concepts that allow us to talk about the possibility and efficiency of such implementation.

PARCS technology concepts, such as: controlled space(CS), recursion and unlimited parallelism can be used in the organization of cloud computing. As described in the introduction, means "cloud computing" characterized by two components: hardware and system tools and service access to users. Note how they can be implemented (or tagged) in terms of PARCS.

Hardware and system tools. Will be presented computers united in local network services running on them, and demons that can be used to implement the structure of CS.

Service access. Some software tool that allows you to run a program PARCS-existing hardware and system tools.

Also, any means to implement cloud computing is *scalability*. Zoom in PARCS-systems implemented because the CS is virtual and can be placed on a single computer and the network (or subnet). And thanks recursion CS can be extended at runtime PARCS program.

Надійшла до редакції 18.10.2016